


EXHIBIT E

US7079752	Azteca America
<p>1. A process for recording, on a recording medium, a scrambled digital video stream, implementing the following steps, in addition to the recording of the scrambled data:</p>	<p>Azteca America supports HLS streaming protocol (“the Standard”). In addition, Azteca America utilizes HLS for delivery of contents to its customers/viewers. As shown below, a video content from Azteca America is streamed and the data traffic is captured showing the media format as HLS, the m3u8 file, (e.g., the Media playlist file comprising links to content chunks in .ts format used by HLS to contain information about the media playing), and the encryption scheme used by the streamed video. In addition, the HLS stream provided through Azteca America provides trick mode operation (such as 10 sec reverse trick mode) to the streamed video.</p> <p>On information and belief, Defendant performs all steps of this claim or, alternatively, to the extent a user performs any step, Defendant conditions the user’s use of the Defendant’s accused instrumentalities using the Standard on the performance of that step as disclosed herein. For example, on information and belief, a user cannot use the accused instrumentality utilizing the Standard as described in this claim chart without performance of the steps recited in this claim. By providing the accused instrumentality utilizing the Standard as disclosed herein, Defendant also controls the manner and/or timing of the functionality described in this claim chart. In other words, for a user to utilize the functionality described in this claim chart, the steps of this claim must be performed in the manner described herein. Without performance of the steps as described herein, the Defendant’s functionality will not be available to users.</p> <p>The Standard practices a process for recording (e.g., recording by means of downloading in a storage), on a recording medium (e.g., a storage mechanism), a scrambled digital video stream (e.g., scrambled video created by making use of AES 128 encryption), implementing the following steps, in addition to the recording of the scrambled data.</p>

miércoles, junio 1, 2022



INMIGRACIÓN

NOTICIAS

LIFESTYLE

ENTRETENIMIENTO


DEPORTES

HOROSCOPO

PROGRAMAS

AZTECAMÁS

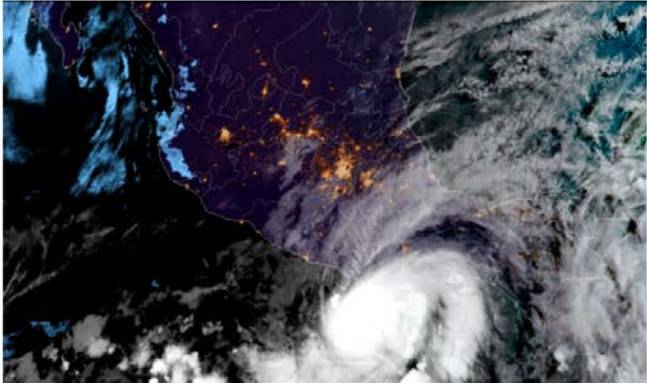
GUÍA TV



BREVE DEL DÍA

Pasión Deportiva: Alan Mozo ya es el nuevo lateral de las Chivas; Benzema, nombrado MVP de la Champions

<https://aztecaamerica.com/>



Lo Último

INMIGRACIÓN MAYO 31, 2022

Protección de indocumentados, casi sin tiempo de ser debatido en Senado

INMIGRACIÓN MAYO 31, 2022

Quiénes son elegibles para tramitar la Green Card: Aquí el paso a paso

LIGA MX MAYO 31, 2022

Atlas: ¿Se avecina un desmantelamiento del blanqueamiento?



<https://aztecaamerica.com/>

Shown below is the URL of .m3u8 master file sent by Azteca America server which identifies the usage of HLS based streaming by Azteca America servers. The m3u8 master file refers to all the variants of the video encoded for various bandwidths and resolutions. The URL of .m3u8 master file is:
<https://cdn.jwplayer.com/manifests/JRsV3ilM.m3u8>

aztecaamerica.com	200	HTTPS	/wp-content/plugins/td-
aztecaamerica.com	200	HTTPS	/wp-includes/js/commen
aztecaamerica.com	200	HTTPS	/wp-content/plugins/wp-
cdn.jwplayer.com	200	HTTPS	/players/x3KDC8S2-Ovg
cdn.jwplayer.com	200	HTTPS	/players/TrrgDfmp-Q9kl
Tunnel to	200	HTTP	ssl.p.jwpcdn.com:443
Tunnel to	200	HTTP	ssl.p.jwpcdn.com:443
Tunnel to	200	HTTP	ssl.p.jwpcdn.com:443
Tunnel to	200	HTTP	ssl.p.jwpcdn.com:443
Tunnel to	200	HTTP	cdn.jwplayer.com:443
Tunnel to	200	HTTP	ssl.p.jwpcdn.com:443
Tunnel to	200	HTTP	ssl.p.jwpcdn.com:443
cdn.jwplayer.com	200	HTTPS	/v2/playlists/x3KDC8S2?
fonts.gstatic.com	200	HTTPS	/s/roboto/v30/KFOlCnqE
fonts.gstatic.com	200	HTTPS	/s/roboto/v30/KFOlCnqE
fonts.gstatic.com	200	HTTPS	/s/opensans/v29/memv
sb.scorecardresearch.com	200	HTTPS	/beacon.js
fonts.gstatic.com	200	HTTPS	/s/roboto/v30/KFOlCnqE
fonts.gstatic.com	200	HTTPS	/s/ffrassans/v16/va9E-4k
fonts.gstatic.com	200	HTTPS	/s/ffrassans/v16/va9E-4k
fonts.gstatic.com	200	HTTPS	/s/merriweather/v30/lu-
fonts.gstatic.com	200	HTTPS	/s/ffrassans/v16/va9E-4k
fonts.gstatic.com	200	HTTPS	/s/roboto/v30/KFOlCnqE
fonts.gstatic.com	200	HTTPS	/s/ffrassans/v16/va9E-4k
fonts.gstatic.com	200	HTTPS	/s/merriweather/v30/lu-
fonts.gstatic.com	200	HTTPS	/s/sourcesanspro/v21/6
fonts.gstatic.com	200	HTTPS	/s/sourcesanspro/v21/6
sb.scorecardresearch.com	302	HTTPS	/b?c1=28c2=33669746
Tunnel to	200	HTTP	content.jwplatform.com
fonts.gstatic.com	200	HTTPS	/s/roboto/v30/KFOlCnqE

Source: Packet capturing by fiddler tool

Get Started
Statistics
Inspectors
AutoResponder
Composer
Fiddler Orchestra

Headers
TextView
SyntaxView
WebForms
HexView
Auth
Cookies
Raw
JSON
XML

Request Headers
[Raw] [Header Definition]

GET /v2/playlists/x3KDC8SZ/recommendations_playlist_id=yqGZfyRv&page_domain=aztecaamerica.com HTTP/1.1

Client
Accept: /*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.62 Safari/537.36

Miscellaneous
Referer: https://aztecaamerica.com/

Security
Origin: https://aztecaamerica.com
sec-ch-ua: "Not A:Brand";v="99", "Chromium";v="102", "Google Chrome";v="102"

Transformer
Headers
TextView
SyntaxView
ImageView
HexView
JSON
WebView
Auth
Caching
Cookies
Raw

JSON
XML

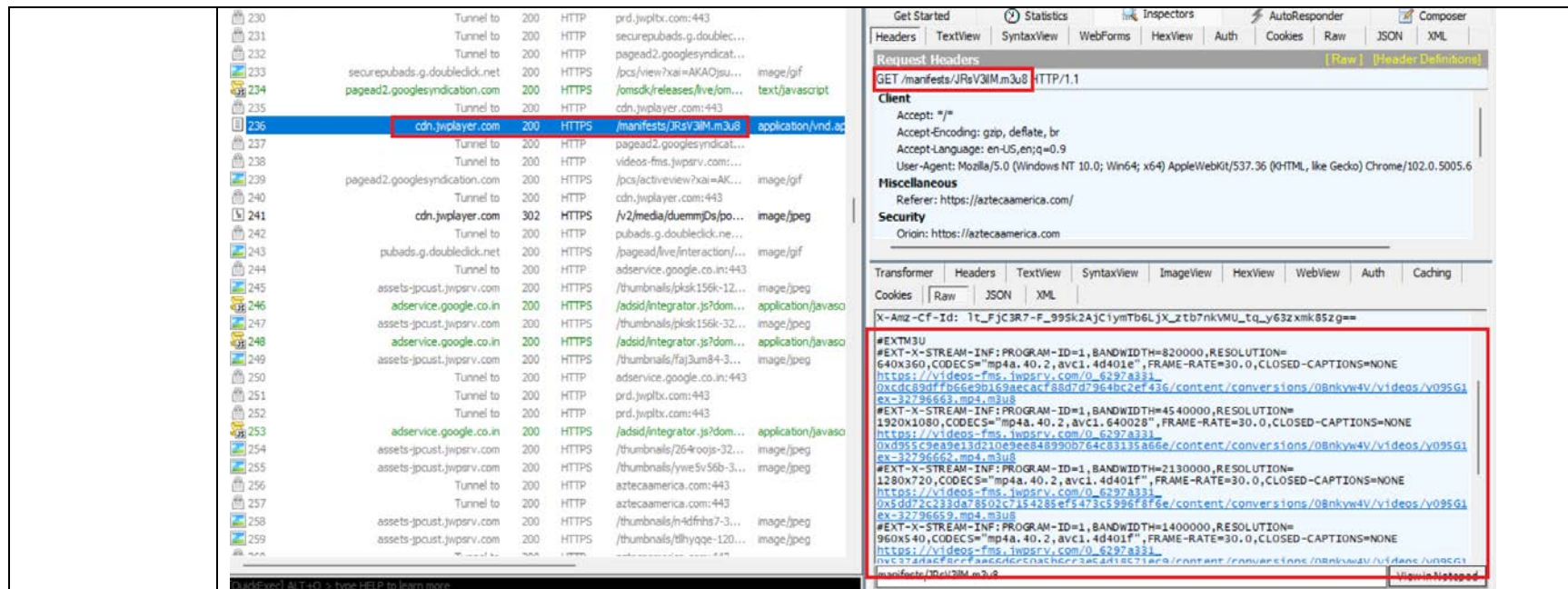
```

480", "width": 480, "type": "image/jpeg"}, {"src": "https://cdn.iwplayer.com/v2/media/JRsv311M/poster.jpg?width=480", "width": 480, "type": "image/jpeg"}, {"src": "https://cdn.iwplayer.com/v2/media/JRsv311M/poster.jpg?width=640", "width": 640, "type": "image/jpeg"}, {"src": "https://cdn.iwplayer.com/v2/media/JRsv311M/poster.jpg?width=720", "width": 720, "type": "image/jpeg"}, {"src": "https://cdn.iwplayer.com/v2/media/JRsv311M/poster.jpg?width=1280", "width": 1280, "type": "image/jpeg"}, {"src": "https://cdn.iwplayer.com/v2/media/JRsv311M/poster.jpg?width=1920", "width": 1920, "type": "image/jpeg"}], {"feedId": "x3KDC8SZ", "duration": 1549, "pubdate": 1621021860, "description": "La 1920, la Sopa de Quindí, Sopas de quindí de pollo con nopales, pipiátallo principal. Supremas de pollo con costra de nuez de la india y salsa de guayabana. Postre: Lassi de guano. U00eIbana con p'u00e9talos de rosa", "tags": "tv azteca, azteca america, azteca us, azteca uno, entretenimiento, uno, comida, Mariano Sandoval, chef, ingredientes, receta, sabores, cocin u00e9dima, seriesId_cocinisma, Supremas de pollo con costra de nuez de la india y salsa de guayabana, seriesId_romeritos, seriesId_bacalao", "recommendations": "https://cdn.iwplayer.com/v2/playlists/yqGZfyRv?media_id=JRsv311M", "sources": [{"file": "https://cdn.iwplayer.com/manifests/JRsv311M.maua" type application/vnd.apple.mpegurl"}, {"file": "https://cdn.iwplayer.com/videos/JRsv311M_538F0CQ3.mp4", type video/mp4, height 180, width 320, label 180p, bitrate 484986, filesize 93905519, framerate 30.0}, {"file": "https://cdn.iwplayer.com/videos/JRsv311M-

```

```
GET https://prd.jwpltx.com/v1/jwplayer6/ping.gif?h=-663665131&e=pa&n=9581568625891688
&abc=0&abt=162_ad-iab-viewability_v4%2C128
_sendDomainToFeedsOn&aid=chud1EzbEeqX9C55nFaE7q&amp=0&ask=nEqvT8RC&at=1&c=1&ccp=0&cp=
0&d=2&eb=0&ed=6&em1=1h2ck2hraygm&i=0&id=JRsv31lM&lid=151wcuw1cufo&lisa=read&mt=1&pbd=1
&pbr=1&pqi=kh41t31uwwks&nph=3&pid=OvgRpnl&pii=0&pl=601&plc=13
&pli=pdcdxh1yb3ud&pp=hlsjs&ppm=VOD&prc=2&ps=4&pss=1&pt=Home%20-%20Azteca%
20America&pu=https://www.aztecaamerica.com%2F&pvr=8.25.1&pyc=0&s=1&sdk=0&stc=1&stpe=
0&t=Supremas%20de%20pollo%20con%20costrak%20de%20nuez%20de%20india%20y%20salsa%
20de%20quayaba&tv=3.39.0&vb=1&vi=0.86&v1=90&wd=1068&ab=1&abid=1ktdur21c19&apid=
1ktdur21c19&cme=0&fed=x3KpC8SZ&fid=7f663e5a-2b2c-4bde-9b2f-47c1450ae847&flc=0
&lng=es&mu=https%3A%2F%2Fcdn.jwplayer.com%2Fmanifests%2FJRsv31lM.m3u8&pd=2
&pdr=&plng=es&pni=0&pogt=Home&pr=7&tb=0&vd=1549&sa=1654061668363 HTTP/1.1
Host: prd.jwpltx.com
Connection: keep-alive
sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="102", "Google Chrome";v="102"
```

Source: Packet capturing by fiddler tool



The screenshot displays the Fiddler network traffic analysis tool. The left pane shows a list of network requests, with a red box highlighting a request to `cdn.jwplayer.com` at `200` status, `HTTPS` protocol, and `/manifests/JRsV3IM.m3u8` path. The right pane shows the details of this request, including the Request Headers and the video manifest content.

Request Headers:

```

GET /manifests/JRsV3IM.m3u8 HTTP/1.1
Client
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.6
Miscellaneous
Referer: https://aztecaamerica.com/
Security
Origin: https://aztecaamerica.com

```

Video Manifest Content:

```

#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=820000,RESOLUTION=
640x360,CODECS="mp4a.40.2,avc1.4d401e",FRAME-RATE=30.0,CLOSED-CAPTIONS=NONE
https://videos-fms.jwpsrv.com/0_6297a331
0xc0d89d0f7b66a9b169aeca788d7d7964bc2ef436/content/conversions/0BnkVw4V/videos/v095G1
0x=32796663,mp4,m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=4540000,RESOLUTION=
1920x1080,CODECS="mp4a.40.2,avc1.640028",FRAME-RATE=30.0,CLOSED-CAPTIONS=NONE
https://videos-fms.jwpsrv.com/0_6297a331
0xd955c9ea9e13d210e9ee848990b764c83135a66e/content/conversions/0BnkVw4V/videos/v095G1
0x=32796663,mp4,m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=2130000,RESOLUTION=
1280x720,CODECS="mp4a.40.2,avc1.4d401f",FRAME-RATE=30.0,CLOSED-CAPTIONS=NONE
https://videos-fms.jwpsrv.com/0_6297a331
0x5d072c233da78502c7154285ef5473c5996f876e/content/conversions/0BnkVw4V/videos/v095G1
0x=32796663,mp4,m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1400000,RESOLUTION=
960x540,CODECS="mp4a.40.2,avc1.4d401f",FRAME-RATE=30.0,CLOSED-CAPTIONS=NONE
https://videos-fms.jwpsrv.com/0_6297a331
0xc0d89d0f7b66a9b169aeca788d7d7964bc2ef436/content/conversions/0BnkVw4V/videos/v095G1
0x=32796663,mp4,m3u8

```

Source: Packet capturing by fiddler tool

Azteca America streams videos with the capability of being played with trick mode (e.g., 10 Sec reverse trick mode)



<https://aztecaamerica.com/>

Azteca America streams scrambled/encrypted content making use of AES 128 encryption.

Transport

Connection: keep-alive

Host: aztecaamerica.com:443

Transformer

Headers

TextView

SyntaxView

ImageView

HexView

WebView

Auth

Caching

Cookies

Raw

JSON

XML

HTTP/1.1 200 Connection Established

FiddlerGateway: Direct

StartTime: 11.04.03.645

Connection: close

Encrypted HTTPS traffic flows through this CONNECT tunnel. HTTPS Decryption is enabled in Fiddler, so decrypted sessions running in this tunnel will be shown in the Web Sessions list.

Secure Protocol: TLS12

Cipher: Aes128 128bits

Hash Algorithm: Sha256 ?bits

Key Exchange: ECDHE_RSA (0xae06) 255bits

Server Certificate: -----

Source: Packet capturing by fiddler tool

An encryption method of AES-128 signals that Media Segments are completely encrypted using the Advanced Encryption Standard (AES) [AES_128] with a 128-bit key, Cipher Block Chaining (CBC), and Public-Key Cryptography Standards #7 (PKCS7) padding [RFC5652]. CBC is restarted on each segment boundary, using either the Initialization Vector (IV) attribute value or the Media Sequence Number as the IV; see [Section 5.2](#).

An encryption method of SAMPLE-AES means that the Media Segments contain media samples, such as audio or video, that are encrypted using the Advanced Encryption Standard [AES_128]. How these media streams are encrypted and encapsulated in a segment depends on the

tos & May

Informational

[Page 15]

8216

HTTP Live Streaming

August 2017

media encoding and the media format of the segment. fMP4 Media Segments are encrypted using the 'cbcs' scheme of Common Encryption [COMMON_ENC]. Encryption of other Media Segment formats containing H.264 [H_264], AAC [ISO_14496], AC-3 [AC_3], and Enhanced AC-3 [AC_3] media streams is described in the HTTP Live Streaming (HLS) Sample Encryption specification [SampleEnc]. The IV attribute MAY be present; see [Section 5.2](#).

<https://tools.ietf.org/html/rfc8216>

1. Introduction to HTTP Live Streaming

HTTP Live Streaming provides a reliable, cost-effective means of delivering continuous and long-form video over the Internet. It allows a receiver to adapt the bit rate of the media to the current network conditions in order to maintain uninterrupted playback at the best possible quality. It supports interstitial content boundaries. It provides a flexible framework for media encryption. It can efficiently offer multiple renditions of the same content, such as audio translations. It offers compatibility with large-scale HTTP caching infrastructure to support delivery to large audiences.

Since the Internet-Draft was first posted in 2009, HTTP Live Streaming has been implemented and deployed by a wide array of content producers, tools vendors, distributors, and device manufacturers. In the subsequent eight years, the protocol has been refined by extensive review and discussion with a variety of media streaming implementors.

The purpose of this document is to facilitate interoperability between HTTP Live Streaming implementations by describing the media transmission protocol. Using this protocol, a client can receive a continuous stream of media from a server for concurrent presentation.

<https://tools.ietf.org/html/rfc8216>

The first line is the format identifier tag #EXTM3U. The line containing #EXT-X-TARGETDURATION says that all Media Segments will be 10 seconds long or less. Then, three Media Segments are declared. The first and second are 9.009 seconds long; the third is 3.003 seconds.

To play this Playlist, the client first downloads it and then downloads and plays each Media Segment declared within it. The client reloads the Playlist as described in this document to discover any added segments. Data SHOULD be carried over HTTP [RFC7230], but, in general, a URI can specify any protocol that can reliably transfer the specified resource on demand.

<https://tools.ietf.org/html/rfc8216>

Playlist files contain URIs, which clients will use to make network requests of arbitrary entities. Clients SHOULD range-check responses to prevent buffer overflows. See also the Security Considerations section of "Uniform Resource Identifier (URI): Generic Syntax" [RFC3986].

<https://tools.ietf.org/html/rfc8216>

An encryption method of AES-128 signals that Media Segments are completely encrypted using the Advanced Encryption Standard (AES) [AES_128] with a 128-bit key, Cipher Block Chaining (CBC), and Public-Key Cryptography Standards #7 (PKCS7) padding [RFC5652]. CBC is restarted on each segment boundary, using either the Initialization Vector (IV) attribute value or the Media Sequence Number as the IV; see [Section 5.2](#).

An encryption method of SAMPLE-AES means that the Media Segments contain media samples, such as audio or video, that are encrypted using the Advanced Encryption Standard [AES_128]. How these media streams are encrypted and encapsulated in a segment depends on the

tos & May

Informational

[Page 15]

8216

HTTP Live Streaming

August 2017

media encoding and the media format of the segment. fMP4 Media Segments are encrypted using the 'cbcs' scheme of Common Encryption [COMMON_ENC]. Encryption of other Media Segment formats containing H.264 [H_264], AAC [ISO_14496], AC-3 [AC_3], and Enhanced AC-3 [AC_3] media streams is described in the HTTP Live Streaming (HLS) Sample Encryption specification [SampleEnc]. The IV attribute MAY be present; see [Section 5.2](#).

<https://tools.ietf.org/html/rfc8216>

RFC 8216

HTTP Live Streaming

August 2017

5. Key Files

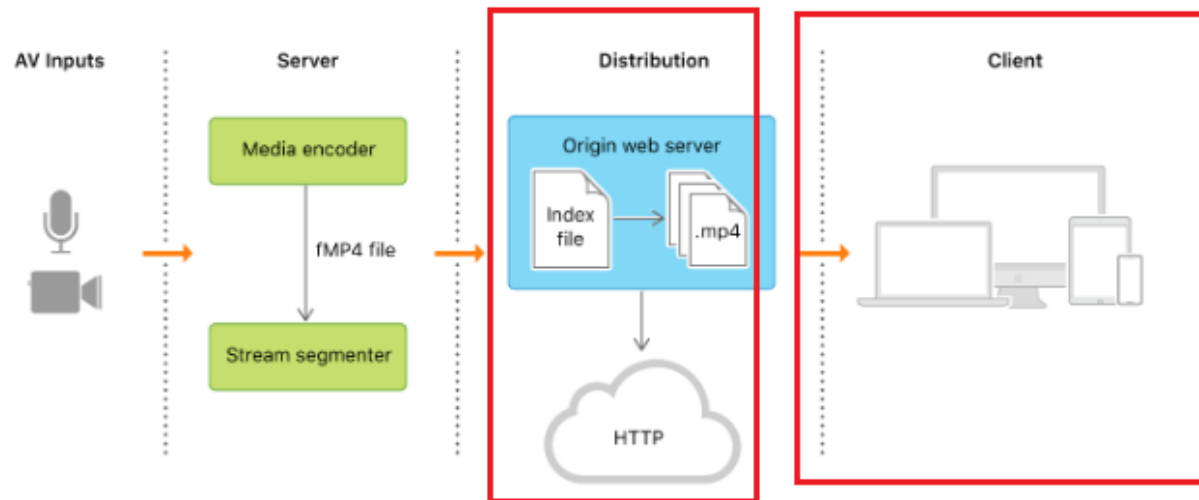
5.1. Structure of Key Files

An EXT-X-KEY tag with a URI attribute identifies a Key file. A Key file contains a cipher key that can decrypt Media Segments in the Playlist.

[AES_128] encryption uses 16-octet keys. If the KEYFORMAT of an EXT-X-KEY tag is "identity", the Key file is a single packed array of 16 octets in binary format.

<https://tools.ietf.org/html/rfc8216>

The following figure shows the components of an HTTP Live Stream.



Apple provides several frameworks that support HTTP Live Streaming, including [AVKit](#), [AVFoundation](#), and [WebKit](#).

https://developer.apple.com/documentation/http_live_streaming

	<p>Latency is cumulative, hence it is added along the whole delivery path from transcoding to the client through the CDN (packaging/origin and caching). Yet, as of today, most of the latency comes from the client: <u>Due to the operation of the protocol (HLS or DASH), and the request/response cycles necessary to obtain the media segments, clients have to maintain a large enough buffer to ensure smooth playback. As an example, an Apple HLS client will start playback once it has buffered at least two segments, resulting in observed latency ranging from 5 to 18 seconds depending on segment durations (2 to 6 seconds).</u></p> <p>To address these issues, both standards have proposed low-latency extensions altering the delivery to the client so that the client can reduce the size of its buffers its buffer sizes:</p> <ul style="list-style-type: none"> ✓ On one side, DASH has built a proposal relying on CMAF combined with HTTP/1.1 chunked transfer encoding to limit the latency induced by the packaging step, with minimal changes on the player side. <p>https://broadpeak.tv/blog/how-apple-hls-is-strengthening-its-hand-in-the-abr-game-with-ll-hls/</p>
<p>descrambling of said scrambled data of said stream so as to extract therefrom additional data corresponding to information required by at least one function of the special mode or</p>	<p>The HLS standard practices descrambling (e.g., decrypting the received encrypted video segments) of said scrambled data of said stream (e.g., scrambled video segments) so as to extract therefrom additional data (e.g., information related to the video segments for trick mode) corresponding to information required by a function of the special mode or “trick mode.”</p>

“trick mode”
(fast forward,
fast rewind,
accelerated
motion, slow
motion, etc.);
and

Transport

Connection: keep-alive

Host: aztecaamerica.com:443

Transformer

Headers

TextView

SyntaxView

ImageView

HexView

WebView

Auth

Caching

Cookies

Raw

JSON

XML

HTTP/1.1 200 Connection Established

FiddlerGateway: Direct

StartTime: 11.04.03.645

Connection: close

Encrypted HTTPS traffic flows through this CONNECT tunnel. HTTPS Decryption is enabled in Fiddler, so decrypted sessions running in this tunnel will be shown in the Web Sessions list.

Secure Protocol: TLS12

Cipher: Aes128 128bits

Hash Algorithm: Sha256 ?bits

Key Exchange: ECDHE_RSA (0xae06) 255bits

Server Certificate: -----

Source: Packet capturing by fiddler tool

Decryption :

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes. Each 128 blocks goes through the 10, 12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

<https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>

Inverse MixColumns :

This step is similar to the MixColumns step in encryption, but differs in the matrix used to carry out the operation.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Inverse SubBytes :

Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

<https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>

GET /players/x3KDC8SZ-OvgRpn1e.js HTTP/1.1

Client

Accept: */*
 Accept-Encoding: gzip, deflate, br
 Accept-Language: en-US,en;q=0.9
 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.62 Safari/537.36

Miscellaneous

Referer: <https://aztecaamerica.com/>

Security

sec-ch-ua: "Not A:Brand":v="99", "Chromium":v="102", "Google Chrome":v="102"

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies

Raw JSON XML

```
technicalError:"This video cannot be played because of a technical error."},
exitFullscreen:"Exit Fullscreen",fullscreen:"Fullscreen",hd:"Quality",liveBroadcast:"
Live",logo:"Logo",mute:"Mute",next:"Next",nextUp:"Next Up",notLive:"Not Live",off:"Off",
pause:"Pause",pipIcon:"Picture in Picture (PiP)",play:"Play",playback:"Play",
playbackRates:"Playback Rates",player:"Video Player",poweredBy:"Powered by",prev:"
Previous",related:{autoplaymessage:"Next up in xx",heading:"More Videos"},replay:"Replay"
rewind:"Rewind 10 Seconds",settings:"Settings",sharing:{copied:"Copied",email:"Email",
embed:"Embed",heading:"Share",link:"Link"},slider:"Seek",stop:"Stop",unmute:"Unmute",
videoInfo:"About This Video",volume:"Volume",volumeSlider:"Volume",shortcuts:{playPause:"
Play/Pause",volumeToggle:"Mute/Unmute",fullscreenToggle:"Fullscreen/Exit Fullscreen",
seekPercent:"Seek %",keyboardShortcuts:"Keyboard Shortcuts",increaseVolume:"Increase
Volume",decreaseVolume:"Decrease Volume",seekForward:"Seek Forward",seekBackward:"Seek
Backward",spacebar:"SPACE",captionsToggle:"Captions On/Off"},captionsStyles:{
```



<https://aztecaamerica.com/>

recording of these additional data on the recording medium.

The product complying with the Standard practices recording of these additional data on the recording medium (e.g., downloading/buffering trick play information to enable trick play mode rendering).

The product must store the data pertaining to trick modes to allow the playback of the video in trick modes.



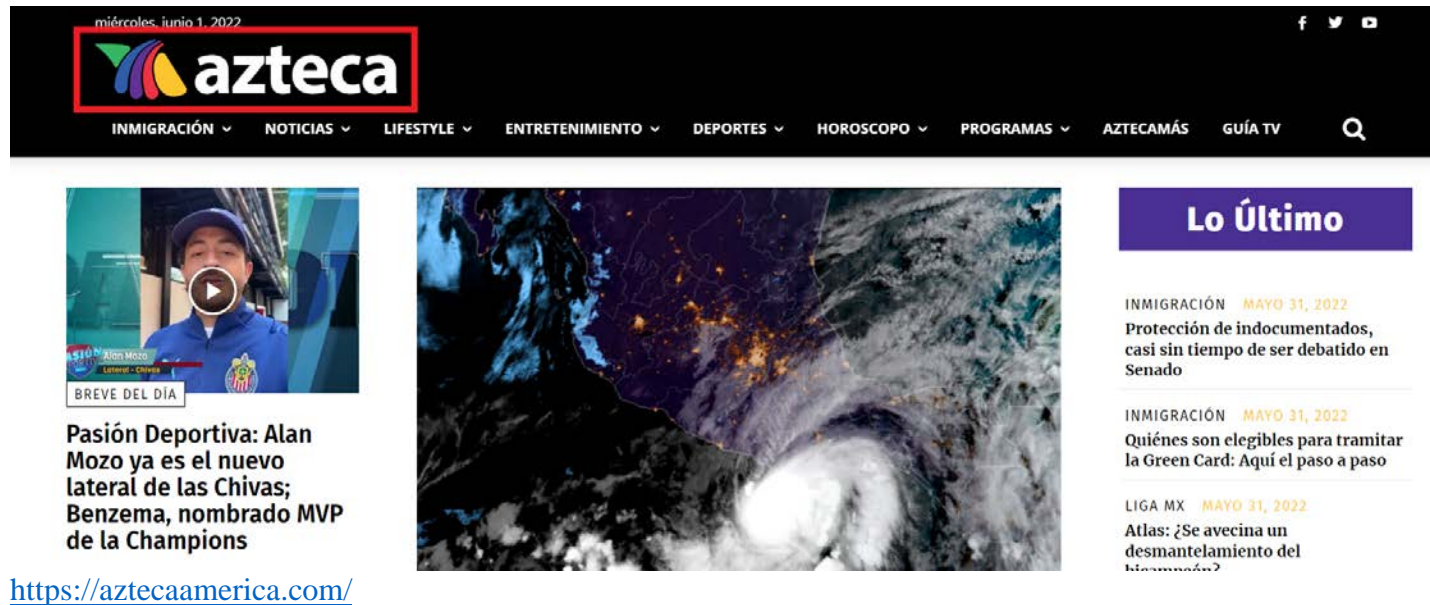
<https://aztecaamerica.com/>

	<p>GET /players/x3KDc8SZ-OvgRpn1e.js HTTP/1.1</p> <p>Client</p> <p>Accept: */*</p> <p>Accept-Encoding: gzip, deflate, br</p> <p>Accept-Language: en-US,en;q=0.9</p> <p>User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.62 Safari/537.36</p> <p>Miscellaneous</p> <p>Referer: https://aztecaamerica.com/</p> <p>Security</p> <p>sec-ch-ua: "Not A:Brand":v="99", "Chromium":v="102", "Google Chrome":v="102"</p> <p>Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies</p> <p>Raw JSON XML</p> <pre> technicalError:"This video cannot be played because of a technical error."}, exitFullscreen:"Exit Fullscreen",fullscreen:"Fullscreen",hd:"Quality",liveBroadcast:" Live",logo:"Logo",mute:"Mute",next:"Next",nextUp:"Next Up",notLive:"Not Live",off:"Off", pause:"Pause",pipIcon:"Picture in Picture (PiP)",play:"Play",playback:"Play", playbackRates:"Playback Rates",player:"Video Player",poweredBy:"Powered by",prev:" Previous",related:{autoplaymessage:"Next up in xx",heading:"More Videos"},replay:"Replay" rewind:"Rewind 10 Seconds",settings:"Settings",sharing:{copied:"Copied",email:"Email", embed:"Embed",heading:"Share",link:"Link"},slider:"Seek",stop:"Stop",unmute:"Unmute", videoInfo:"About This Video",volume:"Volume",volumeSlider:"Volume",shortcuts:{playPause:" Play/Pause",volumeToggle:"Mute/Unmute",fullscreenToggle:"Fullscreen/Exit Fullscreen", seekPercent:"Seek %",keyboardShortcuts:"Keyboard Shortcuts",increaseVolume:"Increase Volume",decreaseVolume:"Decrease Volume",seekForward:"Seek Forward",seekBackward:"Seek Backward",spacebar:"SPACE",captionsToggle:"Captions On/Off"},captionsStyles:{ </pre>
<p>15. A process for decoding a scrambled MPEG stream recorded on a recording medium, for implementing a special mode (“trick mode”),</p>	<p>Azteca America supports HLS streaming protocol (“the Standard”). In addition, Azteca America utilizes HLS for delivery of contents to its customers/viewers. As shown below, a video content from Azteca America is streamed and the data traffic is captured showing the media format as HLS, the m3u8 file, (e.g., the Media playlist file comprising links to content chunks in .ts format used by HLS to contain information about the media playing), and the encryption scheme used by the streamed video. In addition, the HLS stream provided through Azteca America provides trick mode operation (such as 10 sec reverse trick mode) to the streamed video.</p> <p>On information and belief, Defendant performs all steps of this claim or, alternatively, to the extent a user performs any step, Defendant conditions the user’s use of the Defendant’s accused instrumentalities using the Standard on the performance of that step as disclosed herein. For example, on information and belief, a user</p>

comprising the following steps:

cannot use the accused instrumentality utilizing the Standard as described in this claim chart without performance of the steps recited in this claim. By providing the accused instrumentality utilizing the Standard as disclosed herein, Defendant also controls the manner and/or timing of the functionality described in this claim chart. In other words, for a user to utilize the functionality described in this claim chart, the steps of this claim must be performed in the manner described herein. Without performance of the steps as described herein, the Defendant's functionality will not be available to users.

HLS ("the Standard") practices a process for decoding a scrambled MPEG stream (e.g., scrambled video created by making use of AES 128 encryption) recorded on a recording medium (e.g., a buffer for integrated video player on a webpage), for implementing a special mode ("trick mode").





<https://aztecaamerica.com/>

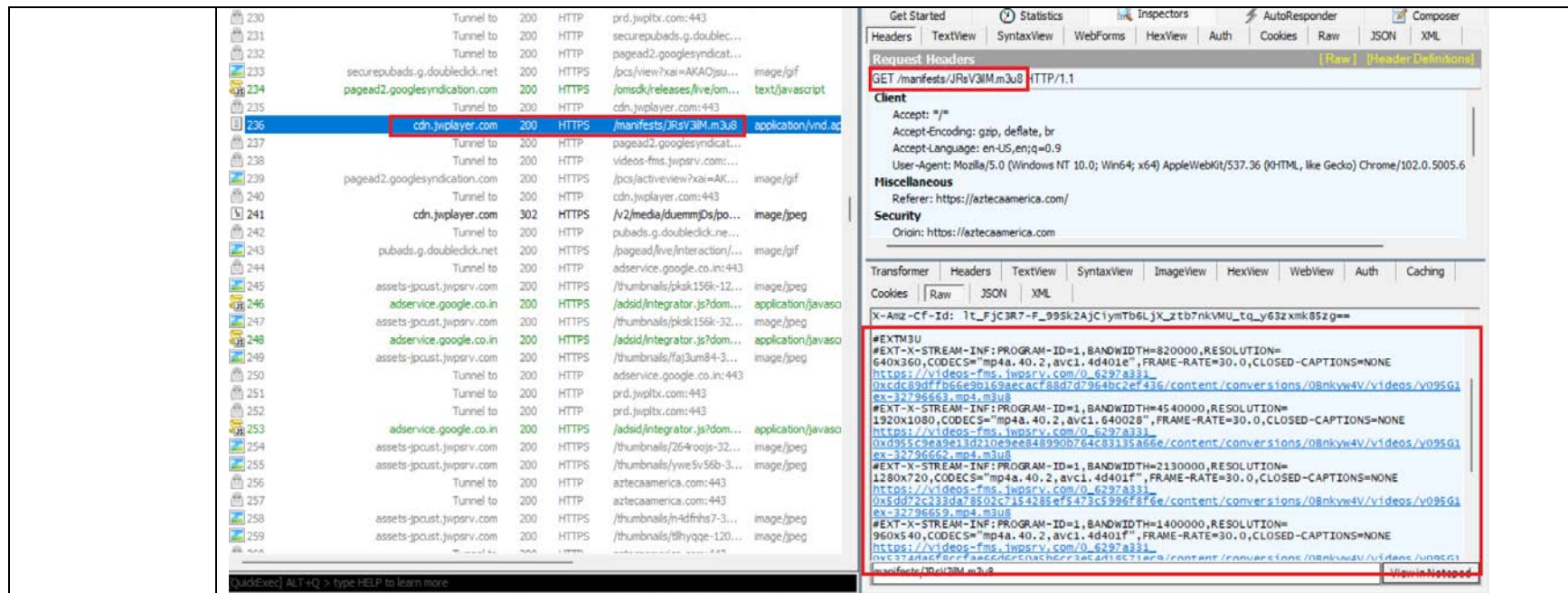
Shown below is the URL of .m3u8 master file sent by Azteca America server which identifies the usage of HLS based streaming by Azteca America servers. The m3u8 master file refers to all the variants of the video encoded for various bandwidths and resolutions. The URL of .m3u8 master file is: [m3u8 file](#)

The screenshot displays the Fiddler tool interface. On the left, a list of network requests is shown. One request from `cdn.jwplayer.com` to `/v2/playlists/x3KDC8SZ?` is highlighted with a red box. The right pane shows the details of this request, including the 'Request Headers' section with a 'GET' method and a 'Referer' from `https://aztecaamerica.com/`. The 'JSON' tab is selected, showing a complex JSON object. Within this object, the 'sources' array contains a video file URL: `https://cdn.jwplayer.com/manifests/20Rsv31M.m3u8`, which is also highlighted with a red box.

Source: Packet capturing by fiddler tool

The screenshot shows a packet capture of a GET request to `https://prd.jwpltx.com/v1/jwplayer6/ping.gif?h=-663665131&e=pa&n=9581568625891688&abc=0&abt=162_ad-iab-viewability_v4%2C128&sendDomainToFeedsOn&aid=chud1EzbEeqX9C55nFaE7q&=0&ask=nEqvt8RC&at=1&c=1&ccp=0&cp=0&d=2&eb=0&ed=6&emi=1h2ck2hraygm&i=0&id=JRsv31M&lid=151wcuwl1cuf0&lsa=read&mt=1&pbd=1&pbr=1&pqi=kh4it31uwyks&ph=3&pid=OvgRpn1e&pii=0&pl=601&plc=13&pli=pcdxdh1yb3ud&pp=nl1sjs&ppm=VOD&prc=2&ps=4&pss=1&pt=Home%20-%20Azteca%20America&pu=https://www.aztecaamerica.com%2F&pv=8.25.1&pyc=0&s=1&sdk=0&stc=1&stpe=0&t=Supremas%20de%20pollo%20con%20costra%20de%20nuez%20de%20la%20india%20y%20salsa%20de%20guayaba&tv=3.39.0&vb=1&vi=0.86&vl=90&wd=1068&ab=1&abid=1ktdur21c19r&apid=1ktdur21c19r&cme=0&fed=x3KDC8SZ&fid=7f663e5a-2b2c-4bde-9b2f-47c1450ae847&flc=0&lng=es&mu=https%3A%2F%2Fcdn.jwplayer.com%2Fmanifests%20JRsv31M.m3u8&pd=2&pdr=es&plng=es&pni=0&poqt=Home&pr=7&tb=1549&sa=1654061668363 HTTP/1.1`. A red box highlights the URL `https://cdn.jwplayer.com/manifests/20Rsv31M.m3u8` within the query parameters.

Source: Packet capturing by fiddler tool



Source: Packet capturing by fiddler tool

Azteca America streams videos with the capability of being played with trick mode (e.g., 10 Sec reverse trick mode)



<https://aztecaamerica.com/>

Azteca America streams scrambled/encrypted content making use of AES 128 encryption.

Transport

Connection: keep-alive

Host: aztecaamerica.com:443

Transformer

Headers

TextView

SyntaxView

ImageView

HexView

WebView

Auth

Caching

Cookies

Raw

JSON

XML

HTTP/1.1 200 Connection Established

FiddlerGateway: Direct

StartTime: 11.04.03.645

Connection: close

Encrypted HTTPS traffic flows through this CONNECT tunnel. HTTPS Decryption is enabled in Fiddler, so decrypted sessions running in this tunnel will be shown in the Web Sessions list.

Secure Protocol: TLS12

Cipher: Aes128 128bits

Hash Algorithm: Sha256 ?bits

Key Exchange: ECDHE_RSA (0xae06) 255bits

Server Certificate: -----

Source: Packet capturing by fiddler tool

An encryption method of AES-128 signals that Media Segments are completely encrypted using the Advanced Encryption Standard (AES) [AES_128] with a 128-bit key, Cipher Block Chaining (CBC), and Public-Key Cryptography Standards #7 (PKCS7) padding [RFC5652]. CBC is restarted on each segment boundary, using either the Initialization Vector (IV) attribute value or the Media Sequence Number as the IV; see [Section 5.2](#).

An encryption method of SAMPLE-AES means that the Media Segments contain media samples, such as audio or video, that are encrypted using the Advanced Encryption Standard [AES_128]. How these media streams are encrypted and encapsulated in a segment depends on the

tos & May

Informational

[Page 15]

8216

HTTP Live Streaming

August 2017

media encoding and the media format of the segment. fMP4 Media Segments are encrypted using the 'cbcs' scheme of Common Encryption [COMMON_ENC]. Encryption of other Media Segment formats containing H.264 [H_264], AAC [ISO_14496], AC-3 [AC_3], and Enhanced AC-3 [AC_3] media streams is described in the HTTP Live Streaming (HLS) Sample Encryption specification [SampleEnc]. The IV attribute MAY be present; see [Section 5.2](#).

<https://tools.ietf.org/html/rfc8216>

1. Introduction to HTTP Live Streaming

HTTP Live Streaming provides a reliable, cost-effective means of delivering continuous and long-form video over the Internet. It allows a receiver to adapt the bit rate of the media to the current network conditions in order to maintain uninterrupted playback at the best possible quality. It supports interstitial content boundaries. It provides a flexible framework for media encryption. It can efficiently offer multiple renditions of the same content, such as audio translations. It offers compatibility with large-scale HTTP caching infrastructure to support delivery to large audiences.

Since the Internet-Draft was first posted in 2009, HTTP Live Streaming has been implemented and deployed by a wide array of content producers, tools vendors, distributors, and device manufacturers. In the subsequent eight years, the protocol has been refined by extensive review and discussion with a variety of media streaming implementors.

The purpose of this document is to facilitate interoperability between HTTP Live Streaming implementations by describing the media transmission protocol. Using this protocol, a client can receive a continuous stream of media from a server for concurrent presentation.

<https://tools.ietf.org/html/rfc8216>

The first line is the format identifier tag #EXTM3U. The line containing #EXT-X-TARGETDURATION says that all Media Segments will be 10 seconds long or less. Then, three Media Segments are declared. The first and second are 9.009 seconds long; the third is 3.003 seconds.

To play this Playlist, the client first downloads it and then downloads and plays each Media Segment declared within it. The client reloads the Playlist as described in this document to discover any added segments. Data SHOULD be carried over HTTP [RFC7230], but, in general, a URI can specify any protocol that can reliably transfer the specified resource on demand.

<https://tools.ietf.org/html/rfc8216>

Playlist files contain URIs, which clients will use to make network requests of arbitrary entities. Clients SHOULD range-check responses to prevent buffer overflows. See also the Security Considerations section of "Uniform Resource Identifier (URI): Generic Syntax" [RFC3986].

<https://tools.ietf.org/html/rfc8216>

An encryption method of AES-128 signals that Media Segments are completely encrypted using the Advanced Encryption Standard (AES) [AES_128] with a 128-bit key, Cipher Block Chaining (CBC), and Public-Key Cryptography Standards #7 (PKCS7) padding [RFC5652]. CBC is restarted on each segment boundary, using either the Initialization Vector (IV) attribute value or the Media Sequence Number as the IV; see [Section 5.2](#).

An encryption method of SAMPLE-AES means that the Media Segments contain media samples, such as audio or video, that are encrypted using the Advanced Encryption Standard [AES_128]. How these media streams are encrypted and encapsulated in a segment depends on the

tos & May

Informational

[Page 15]

8216

HTTP Live Streaming

August 2017

media encoding and the media format of the segment. fMP4 Media Segments are encrypted using the 'cbcs' scheme of Common Encryption [COMMON_ENC]. Encryption of other Media Segment formats containing H.264 [H_264], AAC [ISO_14496], AC-3 [AC_3], and Enhanced AC-3 [AC_3] media streams is described in the HTTP Live Streaming (HLS) Sample Encryption specification [SampleEnc]. The IV attribute MAY be present; see [Section 5.2](#).

<https://tools.ietf.org/html/rfc8216>

RFC 8216

HTTP Live Streaming

August 2017

5. Key Files

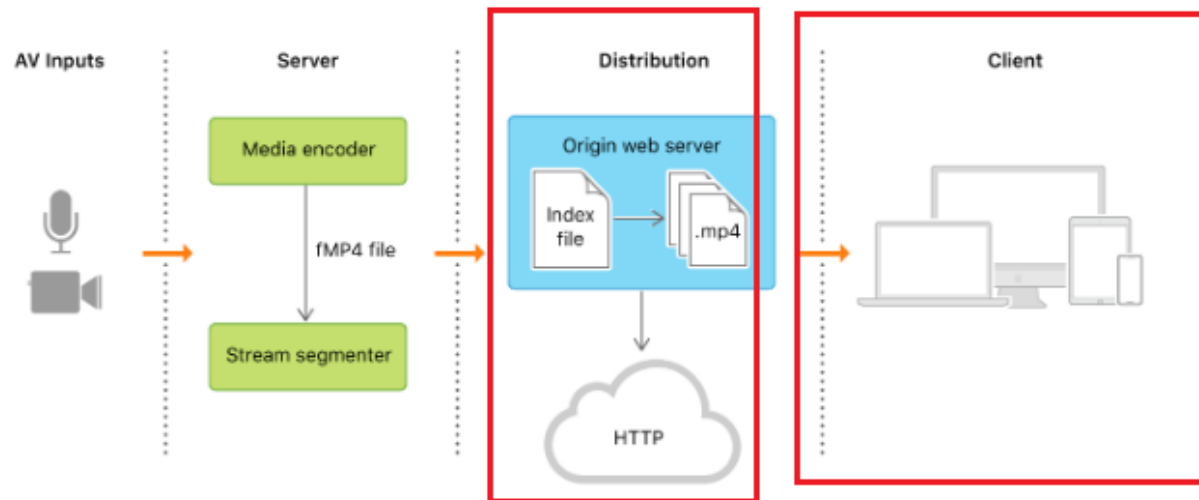
5.1. Structure of Key Files

An EXT-X-KEY tag with a URI attribute identifies a Key file. A Key file contains a cipher key that can decrypt Media Segments in the Playlist.

[AES_128] encryption uses 16-octet keys. If the KEYFORMAT of an EXT-X-KEY tag is "identity", the Key file is a single packed array of 16 octets in binary format.

<https://tools.ietf.org/html/rfc8216>

The following figure shows the components of an HTTP Live Stream.



Apple provides several frameworks that support HTTP Live Streaming, including [AVKit](#), [AVFoundation](#), and [WebKit](#).

https://developer.apple.com/documentation/http_live_streaming

	<p>Latency is cumulative, hence it is added along the whole delivery path from transcoding to the client through the CDN (packaging/origin and caching). Yet, as of today, most of the latency comes from the client: <u>Due to the operation of the protocol (HLS or DASH), and the request/response cycles necessary to obtain the media segments, clients have to maintain a large enough buffer to ensure smooth playback. As an example, an Apple HLS client will start playback once it has buffered at least two segments, resulting in observed latency ranging from 5 to 18 seconds depending on segment durations (2 to 6 seconds).</u></p> <p>To address these issues, both standards have proposed low-latency extensions altering the delivery to the client so that the client can reduce the size of its buffers its buffer sizes:</p> <ul style="list-style-type: none"> ✓ On one side, DASH has built a proposal relying on CMAF combined with HTTP/1.1 chunked transfer encoding to limit the latency induced by the packaging step, with minimal changes on the player side. <p>https://broadpeak.tv/blog/how-apple-hls-is-strengthening-its-hand-in-the-abr-game-with-ll-hls/</p>
<p>reading, from the recording medium, of additional data, other than the scrambled data of the MPEG stream, corresponding to information required by at least one function of the</p>	<p>The accused instrumentality complying with the Standard practices reading, from the recording medium (e.g., a buffer for integrated video player on a webpage), of additional data (e.g., information related to the video segments for trick mode), other than the scrambled data of the MPEG stream (e.g., scrambled video created by making use of AES 128 encryption), corresponding to information required by at least one function of the special mode or “trick mode” (fast forward, fast rewind, accelerated motion, slow motion, etc.).</p> <p>The buffer pertaining to integrated video player on a webpage of the accused instrumentality must store the data pertaining to trick modes to allow the playback of the video in trick modes.</p>

special mode or “trick mode” (fast forward, fast rewind, accelerated motion, slow motion, etc.),

Transport

Connection: keep-alive

Host: aztecaamerica.com:443

Transformer

Headers

TextView

SyntaxView

ImageView

HexView

WebView

Auth

Caching

Cookies

Raw

JSON

XML

HTTP/1.1 200 Connection Established

FiddlerGateway: Direct

StartTime: 11.04.03.645

Connection: close

Encrypted HTTPS traffic flows through this CONNECT tunnel. HTTPS Decryption is enabled in Fiddler, so decrypted sessions running in this tunnel will be shown in the Web Sessions list.

Secure Protocol: TLS12

Cipher: Aes128 128bits

Hash Algorithm: Sha256 ?bits

Key Exchange: ECDHE_RSA (0xae06) 255bits

Server Certificate: -----

Source: Packet capturing by fiddler tool

Decryption :

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes. Each 128 blocks goes through the 10, 12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

<https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>

Inverse MixColumns :

This step is similar to the MixColumns step in encryption, but differs in the matrix used to carry out the operation.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Inverse SubBytes :

Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

<https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>

GET /players/x3KDC8SZ-OvgRpn1e.js HTTP/1.1

Client

Accept: */*
 Accept-Encoding: gzip, deflate, br
 Accept-Language: en-US,en;q=0.9
 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.62 Safari/537.36

Miscellaneous

Referer: <https://aztecaamerica.com/>

Security

sec-ch-ua: "Not A:Brand":v="99", "Chromium":v="102", "Google Chrome":v="102"

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies

Raw JSON XML

```
technicalError:"This video cannot be played because of a technical error."},
exitFullscreen:"Exit Fullscreen",fullscreen:"Fullscreen",hd:"Quality",liveBroadcast:"
Live",logo:"Logo",mute:"Mute",next:"Next",nextUp:"Next Up",notLive:"Not Live",off:"Off",
pause:"Pause",pipIcon:"Picture in Picture (PiP)",play:"Play",playback:"Play",
playbackRates:"Playback Rates",player:"Video Player",poweredBy:"Powered by",prev:"
Previous",related:{autoplaymessage:"Next up in xx",heading:"More Videos"},replay:"Replay"
rewind:"Rewind 10 Seconds",settings:"Settings",sharing:{copied:"Copied",email:"Email",
embed:"Embed",heading:"Share",link:"Link"},slider:"Seek",stop:"Stop",unmute:"Unmute",
videoInfo:"About This Video",volume:"Volume",volumeSlider:"Volume",shortcuts:{playPause:"
Play/Pause",volumeToggle:"Mute/Unmute",fullscreenToggle:"Fullscreen/Exit Fullscreen",
seekPercent:"Seek %",keyboardShortcuts:"Keyboard Shortcuts",increaseVolume:"Increase
Volume",decreaseVolume:"Decrease Volume",seekForward:"Seek Forward",seekBackward:"Seek
Backward",spacebar:"SPACE",captionsToggle:"Captions On/Off"},captionsStyles:{
```



<https://aztecaamerica.com/>

reading, from the recording medium, of scrambled data of the MPEG stream which are determined as a function of the said additional data.

The accused instrumentality complying with the Standard practices reading, from the recording medium (e.g., a buffer for integrated video player on a webpage), of scrambled data of the MPEG stream (e.g., scrambled video created by making use of AES 128 encryption) which are determined as a function of the said additional data (e.g., information related to the video segments for trick mode).

The buffer pertaining to integrated video player on a webpage of the accused instrumentality must store the data pertaining to trick modes to allow the playback of the video in trick modes.



<https://aztecaamerica.com/>

GET /players/x3KDC8SZ-OvgRpn1e.js HTTP/1.1

Client

Accept: */*
 Accept-Encoding: gzip, deflate, br
 Accept-Language: en-US,en;q=0.9
 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.62 Safari/537.36

Miscellaneous

Referer: <https://aztecaamerica.com/>

Security

sec-ch-ua: "Not A:Brand":v="99", "Chromium":v="102", "Google Chrome":v="102"

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies

Raw JSON XML

```
technicalError:"This video cannot be played because of a technical error."},
exitFullscreen:"Exit Fullscreen",fullscreen:"Fullscreen",hd:"Quality",liveBroadcast:"
Live",logo:"Logo",mute:"Mute",next:"Next",nextUp:"Next Up",notLive:"Not Live",off:"Off",
pause:"Pause",pipIcon:"Picture in Picture (PiP)",play:"Play",playback:"Play",
playbackRates:"Playback Rates",player:"Video Player",poweredBy:"Powered by",prev:"
Previous",related:{autoplaymessage:"Next up in xx",heading:"More Videos"},replay:"Replay"
rewind:"Rewind 10 Seconds",settings:"Settings",sharing:{copied:"Copied",email:"Email",
embed:"Embed",heading:"Share",link:"Link"},slider:"Seek",stop:"Stop",unmute:"Unmute",
videoInfo:"About This Video",volume:"Volume",volumeSlider:"Volume",shortcuts:{playPause:"
Play/Pause",volumeToggle:"Mute/Unmute",fullscreenToggle:"Fullscreen/Exit Fullscreen",
seekPercent:"Seek %",keyboardShortcuts:"Keyboard Shortcuts",increaseVolume:"Increase
Volume",decreaseVolume:"Decrease Volume",seekForward:"Seek Forward",seekBackward:"Seek
Backward",spacebar:"SPACE",captionsToggle:"Captions On/Off"},captionsStyles:{
```